

This is a repository copy of *Cross-modal Hashing with Semantic Deep Embedding*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/142022/>

Version: Accepted Version

---

**Article:**

Yan, Cheng, Bai, Xiao, Wang, Shuai et al. (2 more authors) (2019) Cross-modal Hashing with Semantic Deep Embedding. *Neurocomputing*. ISSN 0925-2312

<https://doi.org/10.1016/j.neucom.2019.01.040>

---

**Reuse**

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# Cross-modal Hashing with Semantic Deep Embedding

Cheng Yan<sup>a</sup>, Xiao Bai<sup>a</sup>, Shuai Wang<sup>a</sup>, Jun Zhou<sup>b</sup>, Edwin R. Hancock<sup>a,c</sup>

<sup>a</sup>*School of Computer Science and Engineering and Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China.*

<sup>b</sup>*School of Information and Communication Technology, Griffith University, Nathan, QLD, Australia*

<sup>c</sup>*Department of Computer Science, University of York, Heslington, York, UK*

---

## Abstract

Cross-modal hashing has demonstrated advantages on fast retrieval tasks. It improves the quality of hash coding by exploiting semantic correlation across different modalities. In supervised cross-modal hashing, the learning of hash function relies on the quality of extracted features, for which deep learning models have been adopted to replace the traditional models based on handcraft features. All deep methods, however, have not sufficiently explored semantic correlation of modalities for the hashing process. In this paper, we introduce a novel end-to-end deep cross-modal hashing framework which integrates feature and hash-code learning into the same network. We take both between and within modalities data correlation into consideration, and propose a novel network structure and a loss function with dual semantic supervision for hash learning. This method ensures that the generated binary codes keep the semantic relationship of the original data points. Cross-modal retrieval experiments on commonly used benchmark datasets show that our method yields substantial performance improvement over several state-of-the-art hashing methods.

*Keywords:*

Cross-modal, Deep Hashing, Retrieval, Semantic Embedding,

---

## 1. Introduction

Nearest neighbor (NN) search has been widely adopted in image retrieval. The time complexity of the NN search on a dataset of size  $n$  is  $O(n)$ , which is infeasible for real-time retrieval on large datasets, e.g. multimedia data of large volume and high dimensions. Approximate nearest neighbor (ANN) search makes the NN

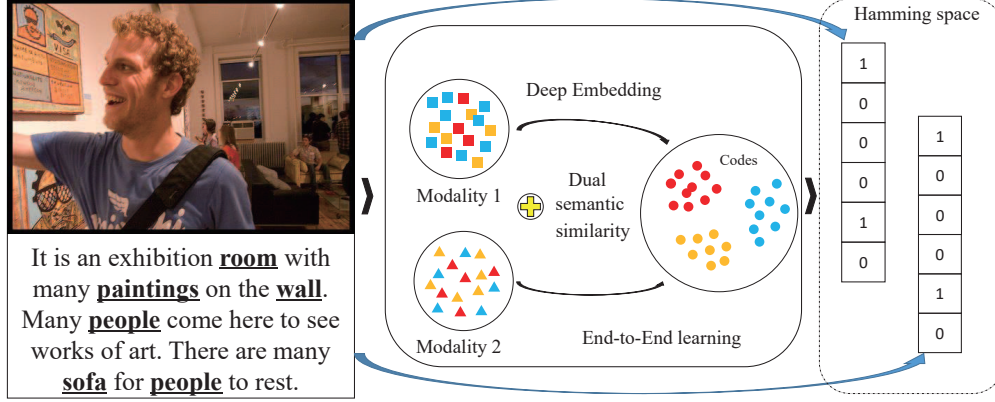


Figure 1: Semantic deep cross-modal hashing for retrieval of images and text sentences.

search scalable, and has become a preferred solution in many computer vision applications [8, 23, 33, 13, 37, 43]. The goal of ANN search is to find approximate results rather than exact ones so as to achieve high speed data processing [28, 14]. Amongst various ANN search techniques, hashing is widely studied because of its efficiency in both storage and speed. By generating binary codes for input data, the retrieval on a dataset with millions of samples can be completed in a constant time using only tens of hash bits [38, 40, 34, 35, 20, 44, 10, 3, 4].

In many applications, data may be collected in more than one modality. For example, in Facebook and Flickr websites, image data are associated with text description or tags. With the rapid growth of such multi-modal data, it is important to properly encode these data for cross-modal retrieval. Given a query in a modality, cross-modal retrieval returns semantically relevant results of another modality. Hashing can be used as a promising solution to handle such retrieval tasks, by transforming high-dimensional cross-modal data into binary codes for fast search [2, 50, 47, 26]. The key in cross-modal hashing is to capture the similarity of data in different modalities. For similar data, the Hamming distance of their corresponding binary codes shall be small.

Cross-modal hashing methods can be divided into two types: unsupervised [18, 36, 46] and supervised [2, 48, 39, 16, 6]. Unsupervised methods do not require labels during the training stage. However, they are faced with a semantic gap, i.e. low-level feature descriptors can not reflect the high-level semantic information of objects and the correlation between cross-modal data is difficult to capture.

Supervised cross-modal hashing methods train binary codes using labels or relevance feedbacks. There is no semantic gap of data, so better hashing quality can be achieved.

Early supervised hashing methods use hand-crafted features to explore shared structures across different modalities [29, 41, 46]. In recent, deep neural networks have been used for feature learning in hashing [2, 19, 21, 49, 24, 27], including in deep cross-modal hashing [16, 6]. In these methods, similarity of samples is only used for feature learning, and the hashing part aims at minimizing the quantization loss from features. It is not difficult to find two gaps. First, the difference between features in different modalities comes from only feature learning process. Second, the difference between features and the corresponding hash codes comes from hashing process. For the hashing process, only minimizing the quantization loss may result in a larger gap between hash codes in different modalities. This means the correlation between samples is lost to some degrees in their final hash codes.

In this paper, we propose a novel Semantic Deep Cross-modal Hashing (SDCH) method, which is an end-to-end deep learning framework. Besides using cross-modal correlation for feature learning, as done by previous works, we also consider dual semantic correlation (correlations between and within modalities) in the loss function for hash learning. The main contributions of this paper are outlined as follows:

- SDCH is a novel end-to-end learning framework which integrates feature learning and hash learning into the same network to guarantee the quality of hash codes.
- We design a loss function with dual semantic supervision and the corresponding network structure to achieve better hashing performance after the semantic hash codes learning.
- We validate the advantages of the proposed method on image-text modalities dataset to show that it outperforms the state-of-the-art methods.

## 2. Related Work

Cross-modal hashing [29, 41, 46] has been an active research topic in computer vision and pattern recognition. Many prior cross-modal hashing methods used unlabeled training data to learn hash functions which transform input data to binary codes. The goal is to preserve the distribution of the original data in

the new Hamming space. Several learning criteria were used, including reconstruction error minimization [12], similarity preservation with graph-based hashing [36, 18], and quantization error minimization [30]. Some cross-modal hashing methods explored supervised information (usually labels) to design hash functions that preserve the relationship of original data, i.e., if two points are similar, their corresponding hash codes from different modalities should be similar. Typical supervised learning frameworks adopted metric learning [5, 25], correlation analysis [42, 48], or neural networks [16, 6]. These methods achieved high accuracy on cross-modal retrieval tasks because supervised information better keeps the cross-modal correlation and reduces the semantic gap in the modelling.

In traditional cross-modal hashing methods, feature extraction step is independent of the hashing process. They adopt shallow architectures and can not well address nonlinearity of data across different modalities. Deep learning based cross-modal hashing methods have been proposed address this problem, [16, 6], however, the correlation is only used for feature extraction but not in the encoding part. Therefore, the learned hash codes can not fully capture the semantic relationship of the original data points.

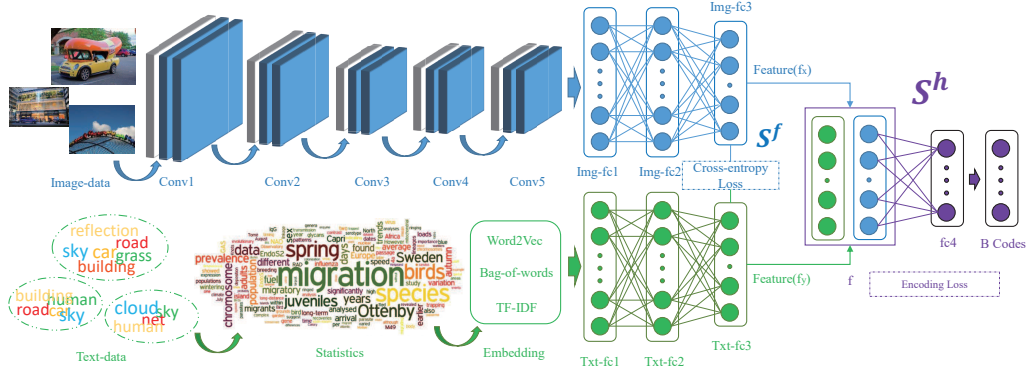


Figure 2: The proposed semantic deep cross-modal hashing method is an end-to-end deep learning framework. We introduce cross and within modal semantic relationship  $S^f$  and  $S^h$  into the learning process, to ensure that the hash codes preserve the semantic correlation of the original data.

### 3. Semantic Deep Cross-modal Hashing

#### 3.1. Model Structure

Our method is an end-to-end deep learning framework with two key parts for cross-modal retrieval. The first part learns the correlation of data in two modalities, and the second part performs semantic hash learning. As shown in Fig. 2, the first part is from *Image-data* and *Text-data* to  $Feature(f_x)$  and  $Feature(f_y)$ . Its goal is to ensure that the outputs of *Img-fc3* and *Txt-fc3* for each sample preserve the correlation between modalities, and passed to the second part for semantic hash learning. Thanks to the end-to-end framework, the loss of the semantic hash learning part also provide feedback to the correlation learning process. Therefore, both learning parts are seamlessly integrated, which ensures that the semantic correlation of each sample can be well preserved by their hash codes.

#### 3.2. Correlation Feature Learning

In the correlation feature learning part, individual pipelines are developed respectively for the image and the text modalities. We adopt the AlexNet [17] for the image network with images resized to  $227 * 227 * 3$  as the input. The last fully connected layer is replaced with a feature layer of  $k$ -dimension ( $k = 256$  in our experiments), so as to reduce the high dimensionality of *fc8*(1000-d) in the original Alexnet for classification. In the text pipeline, each input is a vector with bag-of-words (BOW) representation. The network is composed of three fully connected layers corresponding to the last three layers of the image network with the same number of nodes. Details on these two pipelines are listed in Table 1. The Local Response Normalization (LRN) is used after *img-conv1* and *img-conv2*, and the Rectified Linear Unit (ReLU) is used as an activation function for the first seven layers of the image network and the first two layers of the text network.

Please note that the net structure for image feature extraction is based on widely used model (Alexnet). The main goal of this paper is to design an end-to-end learning framework for cross-modal hashing. It uses deep network for feature learning rather than design different neural networks to extract features. Other deep network structures can also be used for the feature learning task, such as VGG net.

Let  $X = \{x_1, x_2, \dots, x_m\}$  be the input from images, and  $Y = \{y_1, y_2, \dots, y_n\}$  be the input from texts. Let  $\mathbf{f}_{x_i} = f(x_i; \psi_x)$  be the output feature of *Img-fc3* from image  $x_i$  and  $\mathbf{f}_{y_j} = f(y_j; \psi_y)$  be the output feature of *Txt-fc3* from text  $y_j$ , where  $\mathbf{f} \in \mathbb{R}^k$ ,  $\psi_x$  and  $\psi_y$  are the parameters of the two networks respectively. The goal of this part is to guarantee that  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_j}$  capture the correlation of cross-modal

Table 1: Configuration of two networks for images and texts: kernel ( $k$ ), stride ( $s$ ), pad ( $p$ ), pooling kernel ( $pk$ ), and pooling stride ( $ps$ ).

| Layer     | Configuration  |
|-----------|--|
| img-conv1 | $k : 96 \times 11 \times 11, s : 4, p : 0, pk : 3, ps : 2$ |
| img-conv2 | $k : 256 \times 5 \times 5, s : 4, p : 2, pk : 3, ps : 2$  |
| img-conv3 | $k : 384 \times 3 \times 3, s : 0, p : 1$                  |
| img-conv4 | $k : 384 \times 3 \times 3, s : 0, p : 1$                  |
| img-conv5 | $k : 256 \times 3 \times 3, s : 0, p : 1, pk : 3, ps : 2$  |
| fc(img)   | img-fc1:4096    img-fc2:4096<br>img-fc3:256                |
| fc(txt)   | txt-fc1:4096    txt-fc2:4096<br>txt-fc3:256                |

data. To achieve this goal, we define a correlation similarity matrix  $\mathbf{S}^f$  for feature learning, where  $s_{ij}^f = 0$  if image  $x_i$  and text  $y_j$  have different labels and  $s_{ij}^f = 1$  otherwise. Therefore,  $\mathbf{S}^f$  is associated with the semantic information given by labels. Inspired by [7, 16], we use logarithm Maximum a Posteriori (MAP) estimation to learn features  $\mathbf{F}_x = \{\mathbf{f}_{x_1}; \mathbf{f}_{x_2}; \dots; \mathbf{f}_{x_m}\}$  and  $\mathbf{F}_y = \{\mathbf{f}_{y_1}; \mathbf{f}_{y_2}; \dots; \mathbf{f}_{y_n}\}$ . Its objective function is defined as:

$$\log p(\mathbf{F}_x, \mathbf{F}_y | \mathbf{S}^f) \propto \log p(\mathbf{S}^f | \mathbf{F}_x, \mathbf{F}_y) p(\mathbf{F}_x) p(\mathbf{F}_y) \quad (1)$$

where  $p(\mathbf{F}_x)$  and  $p(\mathbf{F}_y)$  are the prior distributions of  $\mathbf{F}_x$  and  $\mathbf{F}_y$ , respectively, and  $p(\mathbf{F}_x, \mathbf{F}_y | \mathbf{S}^f)$  is a likelihood function.

The objective function can be rewritten as:

$$\max \sum_{i,j} \log p(s_{ij}^f | \mathbf{f}_{x_i}, \mathbf{f}_{y_j}) p(\mathbf{f}_{x_i}) p(\mathbf{f}_{y_j}) \quad (2)$$

where  $p(s_{ij}^f | \mathbf{f}_{x_i}, \mathbf{f}_{y_j})$  is the probability of the correlation between  $x_i$  and  $y_j$ . If both  $x_i$  and  $y_j$  are given, it can be calculated as:

$$p(s_{ij}^f | \mathbf{f}_{x_i}, \mathbf{f}_{y_j}) = \phi(\mathbf{f}_{x_i}, \mathbf{f}_{y_j})^{s_{ij}^f} (1 - \phi(\mathbf{f}_{x_i}, \mathbf{f}_{y_j}))^{1-s_{ij}^f} \quad (3)$$

where  $\phi(\mathbf{f}_{x_i}, \mathbf{f}_{y_j}) = 1/(1 + e^{-\mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j}})$  is a sigmoid function.  $\mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j}$  is the inner product of vectors  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_j}$ .

We can consider Eq. (3) as an extension of a logistic regression classifier. If  $s_{ij}^f = 1$ , the larger  $\mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j}$  is, the larger  $p(s_{ij}^f = 1 | \mathbf{f}_{x_i}, \mathbf{f}_{y_j})$  we can get. This means two samples are similar. Conversely, if  $p(s_{ij}^f = 0 | \mathbf{f}_{x_i}, \mathbf{f}_{y_j})$  is large, two samples are dissimilar. When Eq. (3) is maximized, the feature level relationship  $\mathbf{S}^f$  between different modalities can be preserved in the extracted features  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_j}$ . Finally, combining Eqs. (1)-(3), we can get the cross-model loss at feature level:

$$L_f = \sum_{s_{ij}^f} \log(1 + \exp(\mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j})) - s_{ij}^f \mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j} \quad (4)$$

With minimized Eq. (4), if the relationship of two samples is  $s_{ij}^f = 1$ , the inner product of their features shall be large. If  $s_{ij}^f = 0$ , the inner product shall be small. Though the learned features preserve cross-modal correlation in some degrees, directly quantizing them for hash codes generation is not optimal. We design a semantic hash learning part with corresponding constraint to preserve the correlation of binary codes. Integrated in an end-to-end framework, this design also allows the hash learning to contribute to the feature learning, i.e., the gradient in the back-propagation of feature learning network also contains the semantic hash learning part. It is an assurance for high quality hash codes generation.

### 3.3. Semantic Hash Learning

For cross-modal hashing, we aim to encode data from different modalities to ensure their binary codes preserve the correlation of features generated from the original data. Unlike the existing deep cross-model methods which directly quantize the feature, our method uses the learned features for coding with the goal of reducing the coding error. To model the semantic similarity of data, we use class labels to provide the code level relationship for supervised hash function learning. If two samples are in the same class, no matter which modality they belong to, their hash codes should be similar in the Hamming space.

As shown in Fig. 2, the hash codes learning step is from  $Feature(f_x)$  and  $Feature(f_y)$  to the end of the net, which transforms  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_j}$  into binary codes. The proposed network first takes the features  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_j}$  together to form a collection with all features from two modalities. Then we link the feature layer with the hash learning network so it is fully connected with  $fc4$ . The calculation of the final hash codes is based on  $fc4$ .

Let  $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{m+n}\}$  be the hash codes of data samples,  $m$  and  $n$  are the number of data in each modality, and  $\mathbf{S}^h$  denote the pairwise similarity for



hash codes learning. To preserve the semantic similarity, the learned binary codes should be close to  $\mathbf{S}^h$ . Therefore, the binary codes can be learned by minimizing:

$$\begin{aligned} & \left\| \frac{1}{r} \mathbf{B}^\top \mathbf{B} - \mathbf{S}^h \right\|^2 \\ \text{s.t. } & \mathbf{B} \in \{-1, 1\}^{r \times (m+n)} \end{aligned} \quad (5)$$

where  $r$  is the length of the hash codes. Since the value of each element in  $\mathbf{B}$  is binary, the value of each  $\mathbf{S}_{ij}^h$  shall be either 1 or  $-1$ , which means the pairwise relationship is similar or dissimilar. With minimized Eq. (5), if the similarity of two samples  $s_{i,j}^h = 1$ , their hash codes  $\mathbf{b}_i$  and  $\mathbf{b}_j$  are similar. Otherwise,  $s_{i,j}^h = 0$  leads to dissimilar of their hash codes in Hamming space. So Eq. (5) can effectively restrain the hash codes learning. However, solving this objective function is an NP hard problem. We relax the problem by replacing Eq. (5) with:

$$\begin{aligned} \min & \left\| \frac{1}{r} \mathbf{Z}^\top \mathbf{Z} - \mathbf{S}^h \right\|^2 + \|\mathbf{Z} - \mathbf{B}\|^2 \\ \text{s.t. } & \mathbf{B} \in \{-1, 1\}^{r \times (m+n)} \end{aligned} \quad (6)$$

where  $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{m+n}\}$  are the values of  $fc4$  in Fig. (2), and  $\mathbf{B}$  contains the binary codes. We do not directly adopt a symmetric relaxation, such as using  $\mathbf{Z} \approx \mathbf{B}$  and  $\text{sign } \mathbf{Z}$  to obtain  $\mathbf{B}$ , because it may produce a large accumulated quantization error between  $\text{sgn}(\mathbf{Z})$  and  $\mathbf{Z}$ . The new objective function is a discrete optimization function, which is based on the asymmetric relaxation strategy and can further reduce the quantization error. In Eq. (6), we force the binary codes  $\mathbf{B}$  to be similar to the feature, and minimize the differences between the features and matrix  $\mathbf{S}^h$ . In this way, both binary property of the codes and the semantic similarity of data can be guaranteed. Moreover, two terms  $\left\| \frac{1}{r} \mathbf{Z}^\top \mathbf{Z} - \mathbf{I} \right\|^2$  and  $\frac{1}{r} \|\mathbf{Z}\|^2$  are added for independence and balance properties of the hash codes. The final objective function is:

$$\begin{aligned} \min L_h = & \left\| \frac{1}{r} \mathbf{Z}^\top \mathbf{Z} - \mathbf{S}^h \right\|^2 + \beta_1 \|\mathbf{Z} - \mathbf{B}\|^2 \\ & + \beta_2 \left\| \frac{1}{r} \mathbf{Z} \mathbf{Z}^\top - \mathbf{I} \right\|^2 + \frac{\beta_3}{r} \|\mathbf{Z}\|^2 \\ \text{s.t. } & \mathbf{B} \in \{-1, 1\}^{r \times (m+n)} \end{aligned} \quad (7)$$

where  $\beta_i (i \in 1, 2, 3)$  are the hyper-parameters to control the contribution of discrete constraint, independence, and balance properties of codes respectively. Note that, this influences not only the hash learning step, but also the feature learning

part. Eqs. (4) and (7) are very important, especially the Eq. (7), since the feature learning step is partly based on Eq. (7), which is a semantic embedding step to guarantee the quality of hash codes. Since using Eq. (4) only can not guarantee high quality feature learning, we combine Eq. (4) and Eq. (7) to give a dual semantic constraint so as to achieve good performance.

The semantic hash learning objective  $\min L = L_f + \gamma L_h$  can be written as

$$\begin{aligned}
\min_{\Theta_f, \Theta_h, B} L = & \sum_{s_{i,j}^f} \log(1 + \exp(\mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j})) - s_{i,j}^f \mathbf{f}_{x_i}^\top \cdot \mathbf{f}_{y_j} \\
& + \gamma \left( \left\| \frac{1}{r} \mathbf{Z}^\top \mathbf{Z} - \mathbf{S}^h \right\|^2 + \beta_1 \|\mathbf{Z} - \mathbf{B}\|^2 \right. \\
& \left. + \beta_2 \left\| \frac{1}{r} \mathbf{Z} \mathbf{Z}^\top - \mathbf{I} \right\|^2 + \frac{\beta_3}{r} \|\mathbf{Z}\|^2 \right) \\
s.t. \quad & \mathbf{B} \in \{-1, 1\}^{r \times (m+n)}
\end{aligned} \tag{8}$$

where  $\gamma$  is used to adjust the contribution of the feature learning and the hash codes learning parts. In this final objective function, three sets of parameters have to be solved.  $\Theta_f = \{\psi_x, \psi_y\}$  denotes the parameters of the feature learning part, which can be solved based on the final loss  $L$ .  $\Theta_h$  denotes the parameters of hash learning part, whose solution is based on the loss  $L_h$ . Therefore,  $\Theta_f$  is guided not only by the correlation between multi-modal data  $\mathbf{S}^f$  but also by the semantic similarity  $\mathbf{S}^h$ .

An alternating learning strategy is adopted to learn the parameters. We can efficiently optimize the network parameters  $\Theta_f$  and  $\Theta_h$  via automatic differentiation techniques in Google TensorFlow [1]. Specifically, in each iteration, we first optimize  $\mathbf{Z}$  with  $\mathbf{B}$ ,  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_i}$  fixed to obtain the net parameters  $\Theta_h$ . Then we fix  $\mathbf{B}$  and  $\mathbf{Z}$  for optimization of  $\mathbf{f}_{x_i}$  and  $\mathbf{f}_{y_i}$  to obtain the net parameters  $\Theta_f$ . The whole Back-Propagation is accomplished by TensorFlow. Finally with  $\Theta_f$  and  $\Theta_h$  fixed, we can obtain  $\mathbf{f}_{x_i}$ ,  $\mathbf{f}_{y_i}$  and  $\mathbf{Z}$  by Tensorflow Forward-Propagation. After removing the uncorrelated, the target can be written as follows:

$$\begin{aligned}
\max \quad & \text{tr}(\mathbf{Z}^\top \mathbf{B}) = \sum \mathbf{z}_i^\top \mathbf{b}_i \\
s.t. \quad & \mathbf{b}_i \in \{-1, 1\}^r
\end{aligned} \tag{9}$$

It is easy to find that, to maximize the objective function, the hash codes  $\mathbf{b}_i$  for each sample must keep the same sign as  $\mathbf{z}_i$ , so we can get  $\mathbf{B}$  by

$$\mathbf{B} = \text{sign}(\mathbf{Z}) \tag{10}$$

The hash codes of a query can be obtained based on its modality. The hash codes of an image sample can be obtained through the image pipeline, and the hash codes of a text sample can be obtained through the text pipeline.

The pseudo-code for training is shown in Algorithm 1.

---

**Algorithm 1:** The pseudo-code of Semantic Deep Cross-modal Hashing.

---

**Data:** Training image data  $X$  and text data  $Y$ , similarity matrix  $S^f$  and  $S^h$ , the hash codes length  $r$ .

**Result:** Parameters  $\Theta_f$  and  $\Theta_h$  of the network, and the hash codes  $B$ .

Set batch size  $batch = 64$ , the number of iteration  $t = \max(m, n)/batch$

Initialize  $\Theta_f$  and  $\Theta_h$ .

Initialize  $B$  by ITQ [13].

**while**  $epoch \leq maxepoch$  **do**

**for**  $iteration = 1, 2, \dots, t$  **do**

        Randomly sample  $batch$  data from  $X$  and  $Y$  to form mini-batch.

        Do Forward-Propagation to calculate each  $f_{x_i}$ ,  $f_{y_i}$  and  $Z$ .

        Update the parameter  $\Theta_h$  and  $\Theta_f$  in sequence by using automatic Back-Propagation in Google TensorFlow [1].

**end**

    Update  $B$  according to Eq. (10).

**end**

---

## 4. Experiments

Our method was implemented using Google TensorFlow [1]. The network was trained on a NVIDIA TITAN X 12GB GPU. All experiments were undertaken on image-text datasets.

### 4.1. Datasets

Three datasets were used for experiments, including NUS-WIDE [9], MIR-FLICKR [15], and IAPR-TC12 [11].

**NUS-WIDE** is a multi-label dataset containing more than 260k images, with a total number of 5,018 unique tags. Each image is annotated with one or multiple labels from 81 concepts. Following the previous works on the dataset [41, 16],

Table 2: MAPs of different methods for Image-to-Text retrieval task.

| Methods | NUS-WIDE     |              |              | MIR-FLICKR   |              |              | IAPR-TC12    |              |              |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|         | 16bits       | 32bits       | 64bits       | 16bits       | 32bits       | 64bits       | 16bits       | 32bits       | 64bits       |
| IMH     | 0.385        | 0.393        | 0.418        | 0.532        | 0.541        | 0.557        | 0.406        | 0.415        | 0.423        |
| CorrAE  | 0.411        | 0.430        | 0.474        | 0.595        | 0.602        | 0.623        | 0.442        | 0.466        | 0.478        |
| SCM     | 0.431        | 0.445        | 0.465        | 0.573        | 0.589        | 0.606        | 0.537        | 0.569        | 0.591        |
| CM-NN   | 0.601        | 0.616        | 0.623        | 0.680        | 0.731        | 0.740        | 0.542        | 0.548        | 0.465        |
| QCH     | 0.487        | 0.489        | 0.502        | 0.621        | 0.635        | 0.651        | 0.501        | 0.506        | 0.521        |
| SePH    | 0.523        | 0.568        | 0.573        | 0.635        | 0.649        | 0.671        | 0.443        | 0.457        | 0.474        |
| PRDH    | 0.771        | 0.805        | 0.823        | 0.815        | 0.832        | 0.837        | 0.701        | 0.733        | 0.750        |
| DBRC    | 0.463        | 0.471        | 0.481        | 0.581        | 0.596        | 0.602        | 0.451        | 0.467        | 0.478        |
| NDCH    | 0.733        | 0.759        | 0.779        | 0.776        | 0.802        | 0.815        | 0.607        | 0.648        | 0.673        |
| DVSH    | 0.765        | 0.778        | 0.796        | 0.805        | 0.816        | 0.827        | 0.692        | 0.731        | 0.749        |
| DCMH    | 0.773        | 0.804        | 0.815        | 0.805        | 0.824        | 0.835        | 0.625        | 0.697        | 0.723        |
| SDCH    | <b>0.813</b> | <b>0.834</b> | <b>0.841</b> | <b>0.845</b> | <b>0.866</b> | <b>0.873</b> | <b>0.726</b> | <b>0.787</b> | <b>0.803</b> |

we used a subset of 195,834 image-text pairs belonging to 21 most frequent concepts. All images were resized to  $256 * 256 * 3$  and all texts for each sample were represented as bag-of-words (BOW) vectors of 1000 dimensions.

**MIR-FLICKR** is a dataset of  $25k$  images collected from the Flickr website. We selected those samples with at least 20 textual tags for our experiment. All images were resized to  $256 * 256 * 3$  and the corresponding texts were represented as BOW vectors of 1386 dimensions. Each sample was labeled with some of the 24 concepts.

**IAPR-TC12** dataset contains  $20k$  images collected from a wide variety of domains, such as sports, actions, people, animals, cities, landscapes, and so on. Each image is associated with at least one sentence annotation. The text for each data point was represented as a 2912 dimensional bag-of-words vector. All images were resized to  $256 * 256 * 3$ . We used 22 most frequent concepts, and selected the corresponding samples to generate the image-sentence pairs.

For all datasets, if two data samples share at least one common label, we considered them as similar. Otherwise, they were considered to be dissimilar.

Table 3: MAPs of different methods for Text-to-Image retrieval task.

| Methods | NUS-WIDE     |              |              | MIR-FLICKR   |              |              | IAPR-TC12    |              |              |
|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|         | 16bits       | 32bits       | 64bits       | 16bits       | 32bits       | 64bits       | 16bits       | 32bits       | 64bits       |
| IMH     | 0.358        | 0.366        | 0.387        | 0.531        | 0.543        | 0.554        | 0.431        | 0.450        | 0.465        |
| CorrAE  | 0.411        | 0.427        | 0.458        | 0.586        | 0.594        | 0.611        | 0.448        | 0.468        | 0.470        |
| SCM     | 0.433        | 0.448        | 0.465        | 0.536        | 0.544        | 0.571        | 0.515        | 0.528        | 0.531        |
| CM-NN   | 0.585        | 0.597        | 0.623        | 0.670        | 0.681        | 0.709        | 0.497        | 0.505        | 0.522        |
| QCH     | 0.463        | 0.475        | 0.494        | 0.608        | 0.621        | 0.650        | 0.487        | 0.506        | 0.521        |
| SePH    | 0.540        | 0.578        | 0.595        | 0.643        | 0.656        | 0.686        | 0.425        | 0.445        | 0.460        |
| PRDH    | 0.771        | 0.805        | 0.823        | 0.803        | 0.831        | 0.843        | 0.685        | 0.714        | 0.732        |
| DBRC    | 0.453        | 0.469        | 0.471        | 0.583        | 0.596        | 0.601        | 0.463        | 0.479        | 0.491        |
| NDCH    | 0.723        | 0.731        | 0.751        | 0.745        | 0.785        | 0.813        | 0.651        | 0.676        | 0.679        |
| DVSH    | 0.731        | 0.738        | 0.753        | 0.731        | 0.754        | 0.775        | 0.643        | 0.674        | 0.695        |
| DCMH    | 0.770        | 0.803        | 0.811        | 0.801        | 0.821        | 0.833        | 0.679        | 0.707        | 0.726        |
| SDCH    | <b>0.823</b> | <b>0.857</b> | <b>0.868</b> | <b>0.831</b> | <b>0.856</b> | <b>0.863</b> | <b>0.704</b> | <b>0.783</b> | <b>0.797</b> |

#### 4.2. Baselines

For comparison, we used eight state-of-the-art cross-modal hashing methods as baselines, including IMH [36], CorrAE [12], SCM [48], CM-NN [32], QCH [42], SePH [25], PRDH [45], DBRC [22], NDCH [31], DVSH [6], and DCMH [16]. The codes of IMH, CorrAE, CM-NN, SePH, DVSH, DCMH are provided online by the corresponding authors. We implemented the rest of the methods whose codes are not available.

To evaluate the retrieval performance, we followed the approaches in [16, 6, 25, 42] and used three criteria: precision-recall curve, mean Average Precision (mAP) and precision@*top-R* curves. The precision and recall are calculated by

$$\text{precision} = \frac{\text{Number of retrieved relevant pairs}}{\text{Total number of retrieved pairs}} \quad (11)$$

$$\text{recall} = \frac{\text{Number of retrieved relevant pairs}}{\text{Total number of all relevant pairs}} \quad (12)$$

For the mAP, we adopted mAP @*R* = 500 which is the same as in [6, 25, 42].

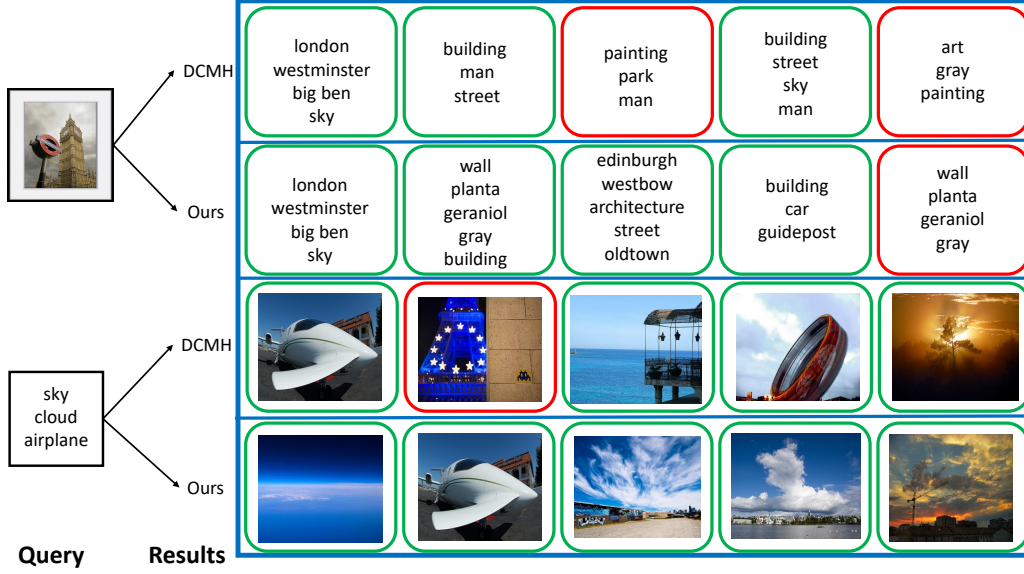
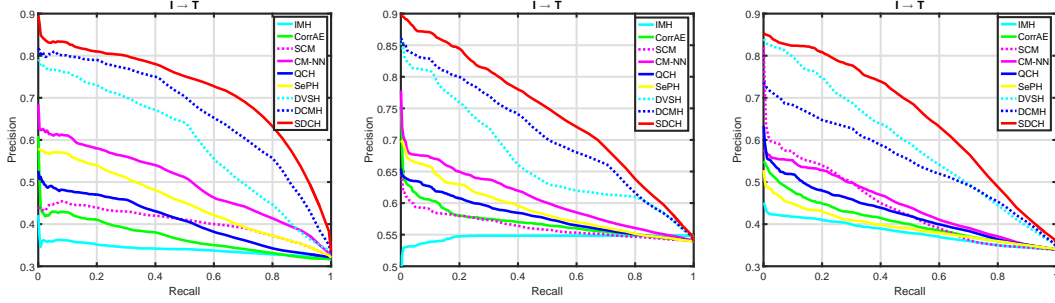


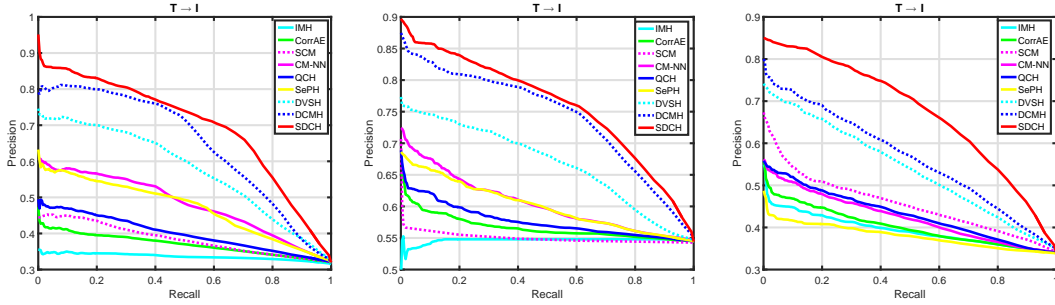
Figure 3: Examples on MIR-FLICKR dataset of the top-5 retrieval results of our SDCH and DCMH whose performance is next to ours. The correct result is shown with a green box.

With respect to hand-crafted feature based methods, for a fair comparison, we used CNN feature with 4096 dimensions extracted from the trained Alexnet [17] to represent each image. For the deep learning baseline methods, we assigned pre-trained parameters to their models, and determine the parameters of each method under comparison by cross-validation, then all results are averaged over five runs.

For our method, we initialized all parameters as follows. For  $\Theta_f$ , the parameters of the first seven layers in the image pipeline were assigned with the values according to the first seven layers of the trained model of the Alexnet [17]. The parameters in pipeline of text ( $Text - fc1$  to  $Text - fc3$ ), the last layer of image pipeline ( $Img - fc8$ ), and  $fc4$  of the hashing part were assigned random values with a normal distribution. In the first epoch, we set the learning rate of  $\Theta_h$  to 0 and only updated  $\Theta_f$ . For  $B$ , we used ITQ [13] with  $z_i (i = 1, \dots, (m + n))$  as the input to give initialize  $B$ . Then we set the learning rate of  $\Theta_f$  to 0 to update  $\Theta_h$  with  $B$  in the second epoch to ensure  $\Theta_h$  have good initial values. After the second epoch, all learning were set to normal in order to train the whole network. The hyper-parameters  $\beta_1, \beta_2, \beta_3$ , and  $\gamma$  were set to 1, 1, 0.1, and 0.01, respectively. The size of mini-batch for training was set to 64, the  $max - epoch$  was set to 50, and the learning rates for the feature extraction part and the hash codes part



P-R curve on NUS-WIDE P-R curve on MIR-FILICKR P-R curve on IAPR-TC12



P-R curve on NUS-WIDE P-R curve on MIR-FILICKR P-R curve on IAPR-TC12

Figure 4: Precision-recall curves on three datasets. The code length is 32. The top row is the results of Image-to-Text retrieval tasks. The bottom row is the results of Text-to-Image retrieval tasks.

were set to  $10^{-3}$  and  $10^{-2}$ , respectively. Since the parameters of the first seven layers in the image pipeline were initialized with the trained model of Alexnet, the corresponding learning rate was set to  $10^{-1}$  of the rate of the feature learning part. All experiments were run for at least five times, and we report the average result.

#### 4.3. Results and Discussions

The mAP results for our SDCH method and other baselines on NUS-WIDE, MIR-FLICKR, and IAPRIAPR-TC12 datasets are reported in Table 2 and Table 3. We evaluated all methods with different lengths of hash codes. Table 2 shows the Image-to-Text retrieval result, which denotes the case where the query is an image and the dataset contains text. Table 3 shows the Text-to-Image retrieval result. We find that SDCH outperforms all the other baselines, especially on the Text-to-Image tasks. SDCH achieves significant increases of 5.7, 3.0, and 7.7 per-

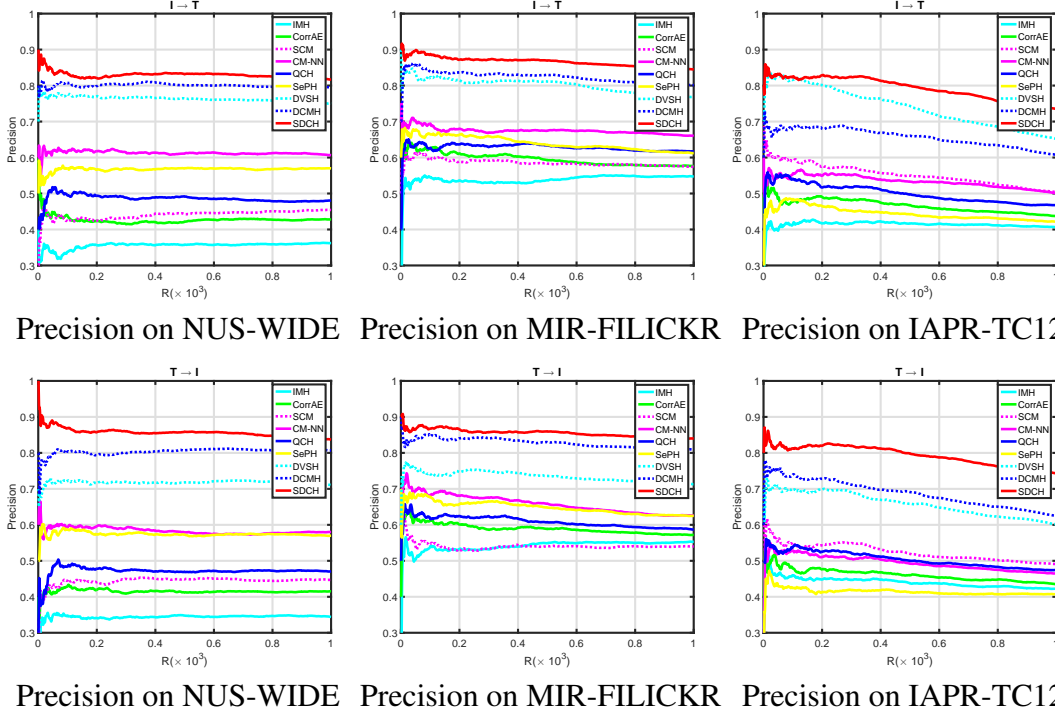


Figure 5: Precision@top-R curves on three datasets. The code length is 32. The top row shows results of Image-to-Text retrieval tasks. The bottom row shows results of Text-to-Image retrieval tasks.

cent in average mAP under 64 bits over the second best method on NUS-WIDE, MIR-FLICKR, and IAPRIAPR-TC12, respectively. The performance of the deep learning-based methods (DVSH, DCMH) is next after ours, which is much better than the hand-crafted feature based methods.

For the hash lookup protocol, the *precision-recall* curves with 32 bits for the Image-to-Text and Text-to-Image tasks on three datasets are shown in Fig. 4. It can be seen that supervised deep learning based methods outperform hand-crafted feature based methods by large margins. Deep learning with a suitable network structure and loss function is essential for improving the performance of cross-model retrieval task. The loss of SDCH is designed not only for the hash codes learning but also for the feature extraction step. Good features are the basis of good hash codes. Moreover, SDCH takes two kinds of data together to keep the relationship in Hamming space for both modalities. With dual semantic supervision and specially designed network structure, SDCH achieves the best



Table 4: The running time of each experiment. Hash time means the time to map one sample to hash code.

| Dataset    | Epoch | Training time | Hash time (image) | Hash time (text) |
|------------|-------|---------------|-------------------|------------------|
| NUS-WIDE   | 50    | 11551s        | 3ms               | 1ms              |
| MIR-FLICKR | 50    | 1539s         | 3ms               | 1ms              |
| IAPR-TC12  | 50    | 1377s         | 3ms               | 1ms              |

cross-modal retrieval performance at all recall levels. From the curves, we can see that SDCH is robust to diverse retrieval scenarios with higher recall tolerating lower precision through the curves.

The *precision@top-R* curves with 32 bits are reported for the two cross-modal retrieval tasks: image query on text dataset ( $I \rightarrow T$ ), and text query on image dataset ( $T \rightarrow I$ ). The results on three datasets are shown in Fig. 5. It can be seen that though using deep features, these hand-crafted feature based methods still have a large gap compared with the deep learning based methods. The curves in these figures shows that SDCH outperforms all other cross-modal retrieval methods, which confirms that SDCH is suitable for the applications that prefer higher precision with fewer top-R retrieved results. The running time of each experiment is shown in Table. 4.

In summary, the experimental results show that the proposed SDCH method has achieved better performance than several state-of-the-art methods in all three hash-based retrieval protocols, especially in the Text-to-Image task. Since our method takes fully advantage of the semantic relationship in the data, including both inter-relationship and inner-relationship among modalities, to supervise the hash codes learning process, the codes of each sample can better preserve the semantic similarity of the original data in the Hamming space of hash codes.

#### 4.4. Parameter Sensitivity

We ran experiments to analyse the influence of hyper-parameters  $\gamma$  and  $\beta_i (i = 1, 2, 3)$ . The range for hyper-parameter  $\gamma$  was set to  $0.001 < \gamma < 2$ . Fig 6(a) shows the MAP results on three datasets with different values, where the code length was 64 bits and  $\beta_i (i = 1, 2, 3)$  are set to 1, 1 and 0.1 respectively.

With  $\gamma \rightarrow 0$ , the model gradually lose the hash learning part, leading to the reduction of performance. With a large  $\gamma$ , the proportion of feature learning part declines which influences the performance of the whole network. The best result is reached when  $\gamma = 0.01$ . With respect to  $\beta_i (i = 1, 2, 3)$ , we changed each of

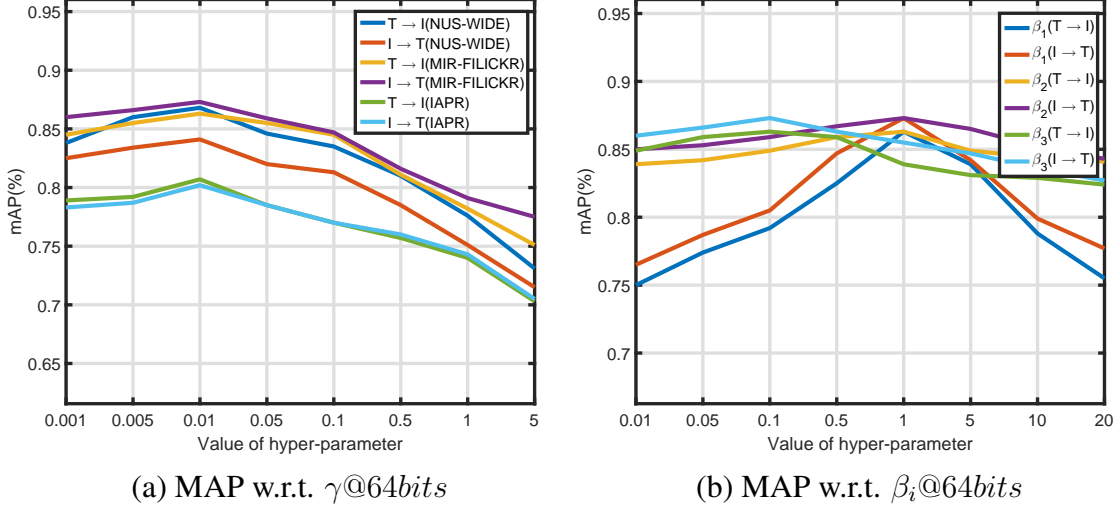


Figure 6: The MAP@64bits versus the parameter  $\gamma \in [0.001, 2]$  and  $\beta_i \in [0.01, 20]$  for the two cross-modal retrieval tasks.

them in the range of  $(0.01, 20)$  while fixing the others to 1 with  $\gamma$  set to 0.01. The results on MIR-FILICKR dataset are shown in Fig. 6(b) with 64 bits code length. It can be seen from the Figure that the  $\beta_1$  is the most important parameter in hashing part since it control the balance of the term restraining the hash codes preserving similarity of each sample. From these figures, we can see that SDCH can outperform all the baseline methods by large margins with the parameters  $\gamma$  between 0.01 and 0.1, and all  $\beta_i$  ( $i = 1, 2, 3$ ) between 0.1 and 1.

## 5. Conclusion

In this paper, we have introduced a novel hashing method, called Semantic Deep Cross-modal Hashing (SDCH), for cross-modal retrieval applications. SDCH is an end-to-end deep learning framework which takes relationship between and within modalities into consideration. A specific loss function with dual semantic supervision and corresponding net structure are designed to guarantee effective hash codes learning. Experiments show that SDCH outperforms several baselines and achieves the state-of-the-art performance on three widely used image-sentences datasets.

## 6. Acknowledgment

This work was supported by the National Natural Science Foundation of China project No. 61772057, and by the Beijing Natural Science Foundation project No. 4162037.

- [1] Martín Abadi and Ashish Agarwal et. al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages III–1247, 2013.
- [3] Xiao Bai, Cheng Yan, Haichuan Yang, Lu Bai, Jun Zhou, and Edwin Robert Hancock. Adaptive hash retrieval with kernel based similarity. *Pattern Recognition*, 75:136–148, 2018.
- [4] Xiao Bai, Haichuan Yang, Jun Zhou, Peng Ren, and Jian Cheng. Data-dependent hashing based on p-stable distribution. *IEEE Transactions on Image Processing*, 23(12):5033–5046, 2014.
- [5] Michael M. Bronstein, Alexander M. Bronstein, Fabrice Michel, and Nikos Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601, 2010.
- [6] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S. Yu. Deep visual-semantic hashing for cross-modal retrieval. In *SIGKDD*, pages 1445–1454, 2016.
- [7] Zhangjie Cao, Mingsheng Long, and Qiang Yang. Transitive hashing network for heterogeneous multimedia retrieval. In *AAAI*.
- [8] M. A Carreira-Perpinan and R Raziperchikolaei. Hashing with binary autoencoders. In *CVPR*, pages 557–566, 2015.
- [9] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, page 48, 2009.
- [10] Kun Ding, Bin Fan, Chunlei Huo, Shiming Xiang, and Chunhong Pan. Cross-modal hashing via rank-order preserving. *IEEE Transactions on Multimedia*, PP(99):1–1, 2017.

- [11] Hugo Jair Escalante, Carlos A. Hernandez, and Jesus A. et. al. Gonzalez. The segmented and annotated iapr tc-12 benchmark. *CVIU*, 114(4):419–428, 2010.
- [12] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. Cross-modal retrieval with correspondence autoencoder. In *MM*, pages 7–16, 2014.
- [13] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *TPAMI*, 35(12):2916–2929, 2013.
- [14] Jae Pil Heo, Youngwoon Lee, Junfeng He, and Shih Fu Chang. Spherical hashing. In *CVPR*, pages 2957–2964, 2012.
- [15] Mark J. Huiskes and Michael S. Lew. The mir flickr retrieval evaluation. In *SIGIR*, pages 39–43, 2008.
- [16] Qing Yuan Jiang and Wu Jun Li. Deep cross-modal hashing. In *CVPR*, pages 3232–3240, 2017.
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [18] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365, 2011.
- [19] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, pages 3270–3278, 2015.
- [20] Kai Li, Guo Jun Qi, Jun Ye, and Kien A. Hua. Linear subspace ranking hashing for cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1825–1838, 2017.
- [21] Wu Jun Li, Sheng Wang, and Wang Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*, pages 1711–1717, 2016.
- [22] Xuelong Li, Di Hu, and Feiping Nie. Deep binary reconstruction for cross-modal hashing. In *IEEE Transactions on Multimedia*, pages 1398–1406, 2017.

- [23] Guosheng Lin, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and David Suter. Fast supervised hashing with decision trees for high-dimensional data. In *CVPR*, pages 1971–1978, 2014.
- [24] Jie Lin, Zechao Li, and Jinhui Tang. Discriminative deep hashing for scalable face image retrieval. In *IJCAI*, pages 2266–2272, 2017.
- [25] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015.
- [26] V Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Cross-modal deep variational hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2017.
- [27] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *CVPR*, pages 2475–2483, 2015.
- [28] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081, 2012.
- [29] Xianglong Liu, Junfeng He, Cheng Deng, and Bo Lang. Collaborative hashing. In *CVPR*, pages 2147–2154, 2014.
- [30] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S Yu. Composite correlation quantization for efficient multimodal retrieval. In *SIGIR*, pages 579–588, 2016.
- [31] Dekui Ma, Jian Liang, Ran He, and Xiangwei Kong. Nonlinear discrete cross-modal hashing for visual-textual data. *IEEE Multimedia*, 24(2):56–65, 2017.
- [32] Jonathan Masci, Michael M. Bronstein, Alexander M. Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *TPAMI*, 36(4):824–830, 2014.
- [33] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and Zhenmin Tang. Inductive hashing on manifolds. In *CVPR*, pages 1562–1569, 2013.

- [34] Fumin Shen, Yan Xu, Li Liu, Yang Yang, Zi Huang, and Heng Tao Shen. Un-supervised deep hashing with similarity-adaptive and discrete optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [35] Fumin Shen, Yang Yang, Li Liu, Wei Liu, Dacheng Tao, and Heng Tao Shen. Asymmetric binary coding for image search. *IEEE Transactions on Multimedia*, 19(9):2022–2032, 2017.
- [36] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, pages 785–796, 2013.
- [37] Christoph Strecha, Alexander M Bronstein, Michael M Bronstein, and Pascal Fua. LDAHash: Improved matching with smaller descriptors. *TPAMI*, 34(1):66–78, 2012.
- [38] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *CVPR*, pages 1–8, 2008.
- [39] Di Wang, Xinbo Gao, Xiumei Wang, and Lihuo He. Semantic topic multi-modal hashing for cross-media retrieval. In *AAAI*, pages 3890–3896, 2015.
- [40] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large-scale search. *TPAMI*, 34(12):2393–2406, 2012.
- [41] Wei Wang, Beng Chin Ooi, Xiaoyan Yang, Dongxiang Zhang, and Yueting Zhuang. Effective multi-modal retrieval based on stacked auto-encoders. pages 649–660, 2014.
- [42] Botong Wu, Qiang Yang, Wei Shi Zheng, Yizhou Wang, and Jingdong Wang. Quantized correlation hashing for fast cross-modal search. In *AAAI*, pages 3946–3952, 2015.
- [43] Bai Xiao, Edwin R Hancock, and Richard C Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.
- [44] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li. Learning discriminative binary codes for large-scale cross-modal retrieval. *IEEE Transactions on Image Processing*, PP(99):1–1, 2017.

- [45] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*, pages 1618–1625, 2017.
- [46] Zhen Yi and Dit Yan Yeung. Co-regularized hashing for multimodal data. In *NIPS*, pages 1376–1384, 2012.
- [47] Dan Zhang, Fei Wang, and Luo Si. Composite hashing with multiple information sources. In *SIGIR*, pages 225–234, 2011.
- [48] Dongqing Zhang and Wu Jun Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, pages 2177–2183, 2014.
- [49] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, pages 2415–2421, 2016.
- [50] Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. Linear cross-modal hashing for efficient multimedia search. In *MM*, pages 143–152, 2013.